COMPUTER

CENTRE

BULLETIN

THIS EDITION

The Department of Computer Science has organized two seminars for third term. These seminars are open to anybody interested in the topics to be discussed. Unfortunately, past seminars have had poor attendance. While the present method of publicising forthcoming seminars through the Bulletin is not entirely satisfactory, it is hoped that in the future it will be possible to remind people of seminar dates a little nearer the time. Meanwhile, any helpful suggestions or criticisms would be welcomed by the Seminar Convenor, Mr J. Williams, ext. 690.

An error in the Operating System and several errors in the Fortran Compiler are reported.

STAFF NEWS

Mrs Noela Kehl has joined the staff as a Data Preparation Assistant. We hope she will enjoy working at the Centre.

SEMINARS IN COMPUTER SCIENCE

Two seminars have been arranged for third term.

The first seminar will be held on Wednesday, 9 September 1970 in Room G13, Engineering Administration Building.

The seminar will commence at 2.00 p.m. and finish no later than 4.00 p.m. The speaker will be Dr B.W. Hicks, Lecturer in the Department of Computer Science, who will discuss *A Student Scheduling System*. The following abstract outlines the theme of the seminar.

As a university increases in size, the logistic problem of designing timetables and assigning students to their respective courses rapidly becomes acute and difficult to handle by manual methods. This is

particularly so in the Faculty of Science, University of Auckland, where, not only have the numbers of students and the variety of subjects increased, but an endeavour is still made to give students an unrestricted choice of course, subject only to counselling and to prerequisites specified for certain units.

In 1968, the Science Timetable Committee recognised two major problems which might be solved by recourse to the digital computer:

1. *The 'streaming' of students upon enrolment into the various duplicate classes available in the existing timetable, with the best possible balance of numbers between classes and the minimum of clashes between different units.*

2. *Revision of the timetable itself so that the first objective can be achieved at optimal level and without prejudicing the principle of unrestricted choice of course.*

The program developed is believed to be an effective solution to the first problem. The second problem seems, for the time being, to be intractable, although a partial solution has been achieved.

The second seminar will be held on Wednesday, 14 October 1970 in Room 214, Engineering Administration Building.

The seminar will commence at 2.00 p.m. and finish no later than 4.00 p.m. The speaker will be Mr R. E. Kelly, Senior Lecturer in the Department of Computer Science, who will discuss *File Organization in Timeshared Systems.*

LIBRARY ACCESSIONS

The following books were acquired by the Libraries of the University of Queensland in June 1970 and may be of interest to readers of the Bulletin.

Wilson, Andrew.   *The bomb and the computer.* 1968. (355.4 WIL Main)

Seminar on the Role of the Computer in the Secondary School. Adelaide, 1969. Proceedings. 1969. (510.78340712 SEM Engin.)

Queensland University. *Computer Centre Executive Committee. Computer Centre development, 1967-69.* 1966. (Q 621.38195 QUE Main)

Svoboda, Antonin. *Computing mechanisms and linkages.* 1964.
(621.8 SVO Elect.)

Bates, Frank. *Programming Language, One.* 1967. (651.8 BAT Engin.)

Bernard, *System 360 COBOL.* 1968. (Qto 651.84 BER Main)
Solomon Martin.

Brooks, *Automatic data processing: System/360 edition.*
Frederick Phillips. 1969. (651.8 BRO Engin.)

International *Introduction to computer systems.* 1969.
Computers Limited. (Qto 651.8 INT T.M.L.)

Kennan, Peter I. *Auditing by computer.* 1969.
(Qto 657.60184 KEN Main)

Malcolme-Lawes, *Programming-Algol.* 1969. (651.8 MAL Engin.)
David J.

Randall, *Systems & procedures for business data processing.*
Clarence Bernard. 1968. (657 RAN Engin.)

Rosenberg, *The computer prophets.* 1969. (651.8 ROS Main)
Jerry Martin.

PDP-10 SYSTEM (FORTRAN)

1. Error in the Fortran Operating System

There is a problem when dealing with free field input from more than one
record. With this mode of input it is essential to have a fixed number of
fields per record although these may be in free field format within the
record.

example:

Values from cards in free field format are read into an array A. The
following coding

```
        READ 1, (A(I), I=1, 16)
    1   FORMAT (1ØF)
```

will read ten values only from the first card and the remaining six values
from the second card. If fewer than ten values are present on the first
card the remaining elements of the array up to the tenth element are
zero filled. If the first card contains more than ten values, those after
the tenth are ignored.

71

2. Error in the Fortran Compiler

(a)   In the following program example, N was used, inadvertently, both as an integer variable and as a function name.

program example:

```
          .
          .
          .
    N = N + 1
          .
          .
          .
    I = N((A + B) + (C + D))
          .
          .
          .
```

Care should be taken to ensure that such errors do not occur since the compiler failed, in this particular case, to detect the error.

This fault in the Fortran Compiler will be corrected in the next version to be released.

(b)   In some cases the value of a DO Loop index is not kept in phase with the internal value for the duration of the DO loop.   In the following example, on the RETURN exit from the subroutine SELECT, L is still set with its original value of 1 regardless of the number of times the DO loop has been performed.

```
          SUBROUTINE SELECT (L)
          COMMON T(1Ø),X
          DO 1Ø L=1,1Ø
          IF (X.LT.T(L)) RETURN
    1Ø    CONTINUE
          STOP
          END
```

The reason for this error is that the compiler, when processing the logical IF statement, fails to realise that the RETURN statement takes program control outside the range of the DO loop. Under these circumstances the DO loop index (L) is not updated and is left at its initial value. The example can be cured by the following rearrangement.

```
                  .
                  .
                  .
         DO 1Ø L=1,1Ø
         IF (X.GT.T(L)) GO TO 1Ø
         RETURN
 1Ø      CONTINUE
                  .
                  .
                  .
```

(c)   The compiler is not protected against invalidly deep parentheses
nests such as in subscripted subscripts. The maximum allowable depth is
not specified but in the example shown a depth of 14 gives a PDL OV
(push down list overflow) error.

example:

```
         DIMENSION I(1Ø)
         K=I(I(I(I(I(I(I(I(I(I(I(I(I(I(J)))))))))))))))
         END
```


3. CREF Error Corrected

An error in the CREF option was reported in Vol. 3 Nos 5 & 6 under
SYSTEM ERRORS. This option now works satisfactorily.

4. File Separator Cards

File Separator Cards are used in BATCH to signify the end of a program or
data file. This card contains the file separator character (card code
11-8-9-4) in column 1. The remainder of the card is ignored and may be
blank, although cards provided by the Computer Centre contain 'FILE
SEPARATOR' in columns 4-17.

In reality, the BATCH program uses the File Separator Card as an indicator
that the next card will be a control card. Thus, although this point was
not mentioned in the article on OVERLAYS in last month's Bulletin, it is
necessary to separate each overlay from the main program and from other
overlays by a File Separator Card. The sequence of cards needed, as
outlined on Page 59, reads:

```
.FORTRAN
        main segment routines
(File Separator Card)

.OVERLAY   AREA=1,  NAME=ANNE                    ; OVERLAY 1
.FORTRAN
        routines to be included in ANNE
(File Separator Card)

.OVERLAY   2   HELEN                             ; OVERLAY 2
.FORTRAN
        routines to be included in HELEN
(File Separator Card)

.OVERLAY   NAME=DIANN   AREA=3                    ; OVERLAY 3

    .
    .
    .
```